app-editors/emacs；臨・兵・闘・者・皆・陣・列・在・前が九字祈祷法なら、Escape - Meta - Alternative - Control - Shift - Super - Front - Hyper - Top は黒客の鍵打秘術

Blurgle Blurgle

version 0.3.0 (since 2009-02-07)

概要

エディタに何を使っても構わないのですが、将来プログラミングをしたい初学者なら少し違います。「エディタなんてどれも同じだと思っていたけど、あの概念を知らなかった過去へは戻れなくなった」が起きるからです。重量級に依って物の見方が変わると言うだけの話で、別に Emacs で紹介する必要は無いのですが、Emacs なのは私の都合 (ルートヴィヒ・ウィトゲンシュタイン氏の言語ゲーム) です。総合開発環境までは触れてません。オフィス・スイートにも触れてません。

- Gentoo 公式サイト: http://www.gentoo.org/
- GentooJP 公式サイト: http://www.gentoo.gr.jp/
- 「優越性と優越性の不在と Gentoo Linux と」: http://dev.gentoo.gr.jp/ yoshino/

目次

1 Missing from /etc/emacs.story 7
1.1 あ、次に進む前に一つ質問です。無粋で申し訳ないですが、Escape - Meta - Alternative - Control - Shift - Super - Front - Hyper - Top って何ですか？ . . . 7

2 Computer Scientists love ed, not just because it comes first alphabetically, but because it is the standard. Everyone else loves ed because it is ED!! 9
2.1 ファイルを開く、ファイルを閉じる . . . 9
2.2 カーソル位置を移動する . . . 9
2.3 文字を挿入する . . . 10
2.4 文字を削除する . . . 11
2.5 取扱説明書 . . . 11

3 Just so you guys know the score. Thank you very much, the king. 12
3.1 Buffer (バッファ) . . . 12
3.2 Register (レジスタ) . . . 13
3.3 Window (ウィンドウ) . . . 14
3.4 Screen (スクリーン) . . . 14
3.5 Frame (フレーム) . . . 15
3.6 Regular Expression (正規表現) での検索 . . . 15
3.7 Regular Expression (正規表現) での置換 . . . 17
3.8 処理 A と処理 B の途中結果を比較したり統合したり . . . 17
3.9 Completion (補完) . . . 18
3.10 *tags (プログラミングに特化した検索機能) . . . 21
3.11 Emacs 独自のスクリプト言語 (Lisp 言語の亜種) . . . 22
3.12 シェルとの連携 . . . 22
3.13 着色 . . . 23
3.14 Template (テンプレート) . . . 24
3.15 Undo (取消) . . . 25
3.16 見出し以外を折り畳み、全体像を把握し易くする . . . 25
3.17 カーソル位置の下のある単語を引金とする機能 . . . 26
3.18 Save (自動保存) . . . 28
3.19 日本語出力 . . . 29
3.20 日本語入力 . . . 30

4 Many people have seen FINE Is Not Emacs, but the one that has its character is THief Isn't Even Fine. 33

5 パッケージ化 40

5.1 「名前を検索」「配布元を検索」「説明文を検索」等 40
5.2 「あのパッケージはどんなコマンドを提供しているのか」「其の USEflag の意味は何か」「此のファイルは何のパッケージに含まれるのか」等 41
5.3 構築作業の自動化 43
5.4 検査作業の自動化 43

付録 A ChangeLog 44

付録 B emacs.sh 45

付録 C Now enough bashing for today, you the king. 57

図目次

表目次

1 Missing from /etc/emacs.story

Emacs は時代を象徴し、良くも悪くも一つの完成形で、此処から別の思想も生まれました。

学生さんから「1983 年生まれって俺より年上じゃん！」と言われそうですが、女性とソフトウェアの歳は気にしない (其の魅力と因果関係は無い) と……じゃなかった、今でも保守どころか新機能が足されるので其れだけ愛好者が居るソフトウェアなのでしょう。

GNU Emacs は GNU is Not Unix シリーズの代表作の一つですから濃いです。しかし此の記事は濃い内容を書く訳でありませんからこんな感じで冗談を書き連ねても構わないのですが、消費されて終わらないものをちょっと意識してみて、十年前の GNU Emacs のマニュアルの表紙には児童書 (西班牙ラ・マンチャ地方の郷士アロンソ・キハーノ) のパロディが採用されていた話にしてみます。世界的な名作の児童書ですが「時代錯誤の変人が頭の弱い従者と漫才みたいな旅をする」ぐらいしか知らない人も居ると思いますし、此処は教科書でもありませんから、付録 E に少し紹介してみます。

1.1 あ、次に進む前に一つ質問です。無粋で申し訳ないですが、Escape - Meta - Alternative - Control - Shift - Super - Front - Hyper - Top って何ですか？

ミレー、ルソー、コロー、ディアズ、ドービニー、ヂュプレ、トロワイヨンのバルビゾン派七星みたいな人の名前ではないです。スピカ、ザビジャ、ポリマ、アウバ、ヘゼ、ヴィンデミアトリックス……の呪文を唱えると乙女座が現れる、そんなノリに似ていますが呪文でなく呪印です。九つの修飾鍵 (Escape key、Meta key、Alternative key、Control key、Shift key、Super key、Front key、Hyper key、Top key) です。私は意図的に Control key と Shift key の二つしか使わないので迫力不足ですが、日常的動作 Kill and Yank (範囲を切り取って別の場所へ貼り付ける)、つまり ‘‘`C-Space <move> C-x b <bufferName> <move> C-y (C-[ y)...`’’ も傍から見れば呪印に見える、だから Emacs (EMACS2FHT) と呼ばれるのでしょうね……と言う冗談です。

知らない方々のために書きますと、実は、GNU Emacs の操作方法は、数十分のチュートリアルぐらいでは大して推測出来無いぐらいのアレです。マニュアルを数時間読むのが早いです。マウスや 3D interaction でなくキーボード主体だから、が理由ではないです。最先端の特殊なマシンの最先端の特殊なキーボードで開発されていた時期が有り、キーボードを転々と変えながら足され続けたのですが、根本的な配置の見直しをしなかった (今更してはいけなかった) のです。

現在編集中の内容を保存する行為 ‘‘`C-x C-w /path/to/file`’’ は ‘‘`Control-x Control-w <path>`’’、つまり「Control key を押しながら x を押す、一度指を離して Control key を押しながら w を押す、再び指を離して対象にしたいファイルのパスを (補完機能を使いながら) 入力する」を意味します。

もう少し詳しく言うと、‘‘`C-x C-w /path/to/file`’’ は関数 “(write-file)” を呼びだしています。↑を押せばカーソル上方移動にしたいのなら関数 “(previous-line)” と結び付けなければなりませんでしたし、Delete key を押せば文字削除にしたいのなら関数 “(delete-char)” と結び付けなければならなかったのです。使用頻度の高い関数 “(write-file)” が ‘‘`C-x C-w /path/to/file`’’ で、使用頻度の低い関数 “(transpose-chars)” が “C-t” なのは修飾鍵が有るキーボードから無いキーボードへ移植された都合です。“(isearch-forward)” が “C-s” (Search の s) と覚え易いのに、“(query-replace)” が “C-[%” と覚え難いのは開発者側の都合です。

上書き (再び結び付ける) して大丈夫です。其の調整は Emacs 独自のスクリプト言語 (Lisp 言語の亜種) で定義する行為を指しますが、後の文章で設定ファイル例 (`~/.emacs.el`、Lisp 言語の亜種) も書き残してみます。

2 Computer Scientists love ed, not just because it comes first alphabetically, but because it is the standard. Everyone else loves ed because it is ED!!

Gentoo Linux の標準は app-editors/nano です。名前は軽量級っぽいですが、実はメールを書くための軽量級 pico よりずっと大きいと言う意味でして、其の通り中量級です。UTF-8 に対する正規表現での置換すら可能です。此れは (慣れると邪魔でしょうけど) 操作ガイドが画面下部に常駐します。

重量級の概念について幾つか書く前に、軽量級の概念も幾つか書いてみます。紙と鉛筆と消しゴムから推測出来る事 (起動方法だけでなく、許されていると期待するもの・禁じられていると推測するものを含む) を実装すればエディタと認めて貰えるでしょうね。

2.1 ファイルを開く、ファイルを閉じる

テキスト編集をするにはファイルを開く機能が必要です。保存する機能も必要です。

.....[code:2.1 ※]

```
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; Open and Close
;;
;; * STANDARD LIBRARY
;;
;; C-x C-f /path/to/file --- find
;; C-x C-w /path/to/file --- write
;; C-x C-c               --- close
;;
```

2.2 カーソル位置を移動する

編集位置へカーソルを移動させる機能が必要です。

.....[code:2.2 ※]

```
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; Move
;;
;; * STANDARD LIBRARY
;;
;; +--           ^
;; |             ^
;; M-a           M-v
;; [SENTENCE]    [UP]
;;
;;               ^
```

```
;;                C-p
;;                [previous]
;;
;; <<          <             >            >>
;; C-a         C-b           C-f          C-e
;; [abc...]    [backward]    [forward]    [end]
;;
;;                V
;;                C-n
;;                [next]
;;
;;                V                          |
;;                V                          --+
;;                C-v                        M-e
;;                [DOWN]                     [SENTENCE]
;;
;; M-x goto-line <number>
;;
;; M-<      ... HOME
;; C-x [    ... positioning
;; M-C-l    ... positioning
;; C-x ]    ... positioning
;; M->      ... END
;;
;; プログラミング支援機能（xxx-mode）では
;; シンタックス単位・ブロック単位での移動機能を追加している。
;; 詳しくは項目参考。
;; ,.,.,.,.,.,.,.....,.,.,.,.,.,.,.,.,.,.,.,..,..,.,.
;; ↑→←↓ Home End PageUp PageDown
(global-set-key [home] 'beginning-of-buffer)
(global-set-key [end] 'end-of-buffer)
(global-set-key [prior] 'scroll-down)
(global-set-key [next] 'scroll-up)
(global-set-key [up] 'previous-line)
(global-set-key [down] 'next-line)
(global-set-key [left] 'backward-char)
(global-set-key [right] 'forward-char)
```

2.3 文字を挿入する

文字列を記述する機能が必要です。

制御文字が何なのか分らなくて自分には関係無いと思われても、そういう文字も計算機が使うのならきっと入力する方法も有るのだろうとは予測出来ます。Emacs は直接記述する手段を持ちますが、解析プログラム側が記号表現 (‘‘¥r’’ は Return を表す等) を用意している場合が有ります。制御文字は見えない文字ですが見えないままだと不便だからです。

日本語を入力するなら ‘‘C-¥’’ ですが、後で紹介します。

.....[code:2.3 ※]

```
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; Force to input
;;
;; * STANDARD LIBRARY
```

```
;;
;; C-q <any_key>
;;
;; C-[   [Escape]
;; M-    [Meta]
;; A-    [Alternative]
;; C-    [Control]
;; S-    [Shift]
;; C-j   [Linefeed]
;; C-m   [Return]
;; C-i   [Tab]
;; C-h   [Backspace]
;; C-d   [Delete]
;;
```

2.4 文字を削除する

文字を削除する機能が必要です。

....[code:2.4 ※]

```
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,..,.,
;; Delete
;;
;; * STANDARD LIBRARY
;;
;; C-h   ... backward
;; C-d   ... forward
;;
(global-set-key "\C-h" 'delete-backward-char)
(global-set-key "\C-d" 'delete-char)
(global-set-key [del] 'delete-backward-char)
(global-set-key [deletechar] 'delete-char)
```

2.5 取扱説明書

全く読む気が無くとも、マニュアルも何処かに有るのだろうとは予測出来ます。

....[code:2.5 ※]

```
;; ;:;:;::::;:;:;:;:;:;:;:;::;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;; Help
;;
;; * STANDARD LIBRARY
;;
;; M-x help-for-help b            ... key -- function
;; M-x help-for-help k <key>      ... function
;; M-x help-for-help w <function> ... key
;; M-x help-for-help v <variable>
;; M-x help-for-help f <function>
;;
```

3 Just so you guys know the score. Thank you very much, the king.

重量級の概念を幾つか紹介してみます。何種類かは他のエディタでも提供しています。一番目立つのは検索や置換などより、テキスト編集したければ起動して終えたら閉じるスタイルから、朝から晩まで起動し続けるスタイルへの習慣そのものの改革でしょう。

どんなものなのか試せる意味で設定内容 (`~/.emacs.el`、Lisp 言語の亜種) も書いてみますが、「此れが最強の設定なんだぜっ！」を紹介するのが主目的では有りません (有名な Dired や Tramp 等を紹介していないし、邪魔な menu bar を消す等の基礎設定が抜けてますから、此れ等だけでは痒い所に手が届かない状態になってしまうかも) ので、興味が有れば書籍で御願いします。

3.1 Buffer (バッファ)

‘‘`C-x C-f <path>`’’ を何回か繰り返すと複数のファイルが同時に開けます。各ファイルは別々のバッファに格納され区別されますが、バッファ間の連携が出来ます。Kill and Yank (範囲を切り取って別の場所へ貼り付ける) は ‘‘`C-Space <move>`’’ で範囲切り取り、‘‘`C-x b <bufferName>`’’ でバッファ移動、‘‘`<move>`’’ でカーソル位置の移動、‘‘`C-y (C-[ y)...`’’ で貼り付けです。

どういう事が起きるのかと言えば、終了させる事無く朝から晩まで同じ Emacs プロセスで十個二十個開き続けられます。デタッチ・アタッチの概念もしくはサーバ・クライアントの概念を加えれば、ログアウトして別マシンからでも前回環境 (十個二十個の編集途中) から続けられます。一種のデーモンになります。

余談ですけど大きなファイルは一つの大きなバッファになりますが、一つのバッファを手頃な大きさへ分割する事を Narrowing (ナローイング) と呼びます。

以下はバッファ操作を (達人が好みがちな) 結果だけを表示するのではなく、手間が増えますけど途中に一覧を挟む設定です。‘‘`C-x b <bufferName>`’’ で移動出来ますが、‘‘`C-x C-b`’’ でバッファ一覧を経由しても移動出来ます。

.....[code:3.1 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
(global-set-key "\C-x\C-b" 'bs-show)
(setq bs-attributes-list
      '(
        ("-" 1 1 left bs--get-marked-string)
        ("M" 1 1 left bs--get-modified-string)
        ("R" 2 2 left bs--get-readonly-string)
        ("Coding" 15 15 left bs--get-coding-string)
        ("Buffer" 20 20 left bs--get-name)
```

```
        ("Size" 8 8 right bs--get-size-string)
        ("" 1 1 left " ")
        ("Mode" 15 15 left bs--get-mode-name)
        ("" 1 1 left " ")
        ("File" 10 60 left bs--get-file-name)
        )
)
(defun bs--get-coding-string (start-buffer all-buffers)
  (if (coding-system-get buffer-file-coding-system 'mime-charset)
      (format "%s" (coding-system-get buffer-file-coding-system 'mime-charset))
    "undefined"
  )
)
```

3.2 Register (レジスタ)

エディタはユーザが明示的に作るバッファの他にレジスタと呼ばれる内部処理で利用する領域を保持しています。例えば Kill (切り取り) と Yank (貼り付け) が其れです。

以下はレジスタ操作を (達人が好みがちな) 結果だけを表示するのではなく、手間が増えますけど途中に一覧を挟む設定です。貼り付け動作を ‘‘C-y’’ の代わりに ‘‘C-x C-y’’ とすると複製・切取の履歴一覧から選択出来ます。

.....[code:3.2 ※]

```
;;; emacs-23
;;; * app-emacs/browse-kill-ring
;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Kill (切り取り) と Yank (貼り付け)
;;
;; C-k                 ... Kill
;; <region> C-w        ... Cut
;; <region> M-w        ... Copy
;;
;; C-y                 ... Yank
;;   Prev:
;;   Next: M-y
;; C-x C-y
;;   Prev: p
;;   Next: n
;;   Yank: y
;;   Undo: U
;;   Edit: e
;;   Return: C-c C-c
;;
(require 'browse-kill-ring)
(global-set-key "\C-x\C-y" 'browse-kill-ring)
(setq browse-kill-ring-display-style 'one-line
      browse-kill-ring-quit-action 'kill-and-delete-window
      browse-kill-ring-resize-window t
      browse-kill-ring-separator "----------"
      browse-kill-ring-highlight-current-entry t
      browse-kill-ring-separator-face 'region
      browse-kill-ring-maximum-display-length 100
)
```

3.3 Window (ウィンドウ)

画面分割を行うと画面は複数のウィンドウで構成される様になります。見栄えの問題です。

‘‘C-x 3’’ は右半分と左半分、‘‘C-x 2’’ は上半分と下半分に分割です。’‘‘C-x o’’ でカーソルがウィンドウ間を移動します。画面を戻すなら ‘‘C-x 0’’ もしくは ‘‘C-x 1’’ です。

....[code:3.3 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
; N/A
```

3.4 Screen (スクリーン)

複数のウィンドウから構成された画面を単位とします。バッファ群はスクリーン間で共通ですから作業のグループ化と言えます。学問的意味は違いますが、表面的効果は Xwindow 上の Window Manager と同じです。

実は標準パッケージでありません。マージして下さい。サーバ・クライアントの概念もしくはデタッチ・アタッチの概念を齎す ELisp 拡張パッケージもしくは外部プログラムが有れば必要無いかも知れません。『app-misc/tmux ; 仮想端末の集合体を複数の物理端末で共有する』に tmux.sh を載せています。

....[code:3.4 ※]

```
;;; emacs-23
;;; * app-emacs/elscreen
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;; C-z C-c   ... create
;; C-z C-k   ... kill
;; C-z 0     ... move
;; C-z 9     ... move
;; C-z w     ... list
;; C-z ?     ... help
;;
(require 'elscreen)
```

3.5 Frame (フレーム)

既に起動している Emacs プロセスと資源共有する形でもう一つ起動させると、一つの Emacs プロセスがフレーム (Frame) を二つ所有していると表現されます。学問的意味は違いますが、表面的効果は Xwindow 上の Window Manager と同じです。

‘‘C-x 5 f’’ でフレームを作り、‘‘C-x 5 0’’ で消‘‘C-x 5 b’’ でカーソルがフレーム間を移動します。サーバ・クライアントの概念もしくはデタッチ・アタッチの概念を齎す ELisp 拡張パッケージもしくは外部プログラムが有れば必要無いかも知れません。『app-misc/tmux ; 仮想端末の集合体を複数の物理端末で共有する』に tmux.sh を載せています。

.....[code:3.5 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
; N/A
```

3.6 Regular Expression (正規表現) での検索

‘‘C-[C-s’’ が正規表現での検索です。専門書には索引が付きますが、検索とは直接カーソル位置が候補の位置へ移動します。

ディレクトリ単位で検索するのなら、一つ一つ実行するよりも ELisp 拡張パッケージを足すか外部プログラムを使うかが無難です。

以下は検索を (達人が好みがちな) 結果だけを表示するのではなく、手間が増えますけど途中に一覧を挟む設定です。検索動作を ‘‘C-[C-s <search>’’ の代わりに ‘‘C-c C-s <search>’’ とすると候補一覧から選択出来ます。

.....[code:3.6 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;;
;; M-C-r <search>
;; M-C-s <search>
;;   Prev: C-r
;;   Next: C-s
;;
(setq seach-highlight t
      query-replace-highlight t
      case-fold-search t
```

```
      case-replace t
)


;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; 検索 (一覧経由)
;;
;; * STANDARD LIBRARY
;;
;; M-x occur <search>     --- one
;; M-x moccur <search>    --- all
;; C-c C-s <search>
;;   Prev: M-p
;;   Next: M-n
;;   Show: C-c C-f <move>
;;   Go:   C-c C-c
;;
(setq list-matching-lines-face 'region
      list-matching-lines-default-context-lines 2
)
(global-set-key "\C-c\C-s" 'moccur)


;; ,.,.,...,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; * color-moccur
;; * app-emacs/color-moccur
;; * http://www.bookshelf.jp/
;;
;; M-x moccur <search>
;;   Prev: M-p
;;   Next: M-n
;;   Prev: p
;;   Next: n
;;   Search: s <search>
;;   Undo: /
;;   Show: t <move>
;;   Go:   C-c C-c
;;   Edit: r          ... read only >>> writable
;;     Cancel: <region> C-c C-r
;;     Exit: C-x C-s
;; M-x doccur <directory> <search>
;; M-x clean-dmoccur-buffers
;; <dired> m... M-x dired-do-moccur <search>
;;
(require 'color-moccur)
(if (load "color-moccur" t)
    (progn
      (setq moccur-following-mode-toggle t
            moccur-grep-following-mode-toggle t
            moccur-kill-moccur-buffer t
            moccur-kill-buffer-after-goto t
            kill-buffer-after-dered-do-moccur t
            moccur-use-migemo nil
            moccur-use-ee nil
            color-moccur-default-ime-status t
            moccur-maxmum-displayed-with-color 100
            ;; kiB
            dmoccur-maximum-size 1024
            dmoccur-recursive-search t
      )
    )
    (message "Failure color-moccur")
)
```

3.7 Regular Expression (正規表現) での置換

‘‘C-[C-%’’ が正規表現での置換です。検索して修正して、検索して修正して、検索して修正して……を自動化させたものです。

ディレクトリ単位で置換するのなら、一つ一つ実行するよりも ELisp 拡張パッケージを足すか外部プログラムを使うかが無難です。

以下は検索を (達人が好みがちな) 結果だけを表示するのではなく、手間が増えますけど途中に一覧を挟む設定です。置換動作を ‘‘C-[C-% <search> <replace>’’ の代わりに ‘‘C-c C-s <search> r’’ とすると候補一覧から選択出来ます。

.....[code:3.7 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;;
;; M-C-% <search> <replace>
;;   Yes:    y
;;   No:     n
;;   Edit:   C-r
;;   Return: M-C-r
;;
(global-set-key "\M-C-%" 'query-replace-regexp)


;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; 置換 (一覧経由)
;;
;; * moccur-edit
;; * app-emacs/moccur-edit
;; * http://www.bookshelf.jp/
;;
;; See color-moccur
;;
(require 'moccur-edit)
(setq moccur-edit-highlight-edited-text t
      moccur-edit-remove-overlays t
      moccur-edit-remove-overlays-after-save-buffer t
      moccur-query-when-buffer-read-only t
)
```

3.8 処理 A と処理 B の途中結果を比較したり統合したり

似ている二つのファイルを管理します。

此の概念を Emacs はサポートしますが、ヴァージョン管理システムと呼ばれる外部プログラムに任せた方

が良いかも知れません。

.....[code:3.8 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;; M-x ediff-files <file> <file>
;; M-x ediff-buffers <buffer> <buffer>
;; M-x ediff-directories <directory> <directory>
;; M-x ediff-files3 <file> <file> <file>
;; M-x ediff-buffers3 <buffer> <buffer> <buffer>
;; M-x ediff-directories3 <directory> <directory> <directory>
;;
;;   V
;; <   >
;;   v
;;
;; |
;; h
;; p/n
;; a
;; b
;; r a
;; r b
;; w d <save_patch>
;;
;; M-x ediff-patch-file <patch> <file>
;;   file   >>> <file> + <patch>
;;   .patch >>> <patch>
;;   .orig  >>> <file>
;; ,.,.,.,..,.,.,.,.,.,.,,.,.,.,.,.,.,.,.,,,,,,..,.,.
;; M-x ediff-merge-files <file> <file>
;;
;;   V
;; <   >
;;   v
;;
;; |
;; h
;; p/n
;; a
;; b
;; r
;; w c <save_file>
;;
(setq ediff-window-setup-function 'ediff-setup-windows-plain)
(set-default 'ediff-auto-refine-limit 10240)
```

3.9 Completion (補完)

補完とは、ユーザが何か入力しなければならない場合、其れが有限の候補から選べるものならば、今までの入力値からユーザの意図を推測する (結果としてキーボードを叩く回数が減る) 機能です。不完全な英単語やファイル名に限らずバッファ上の識別子まで完全な形を推論し候補を提案してくれます。

同じくキーボードを叩く回数を減らすには履歴から選ぶ方法も有ります。

.....[code:3.9 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
(icomplete-mode t)
(minibuffer-electric-default-mode t)


;; ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;; * mcomplete
;; * app-emacs/mcomplete
;; * http://homepage1.nifty.com/bmonkey/emacs/elisp/
;;
;; Tab...
;;   prefix:    C-p
;;   substring: C-n
;(require 'mcomplete)
;(turn-on-mcomplete-mode)


;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; History
;;
;; * STANDARD LIBRARY
;;
;; M-p
;; M-n
;; M-r <search>
;; M-s <search>
;;   Prev: M-r
;;   Next: M-s


;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; ファイル指定時の補完機能強化
;;
;; * STANDARD LIBRARY
;;
;; C-x C-f <filename> (C-x C-i)...    --- Tab
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
(require 'filecache)
(file-cache-add-directory-list (list
                                "~/bin/Bash/"
                                "~/var/GentooLinux/local/articles/"
                                "~/etc/"
                                "/etc/"
                                "/usr/include/"
                                )
)
(define-key minibuffer-local-completion-map "\C-x\C-i" 'file-cache-minibuffer-complete)


;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;; 登録単語又はバッファ内からカーソル位置の文字列を補完する
;;
;; * STANDARD LIBRARY
;;
;; <region> C-u 0 C-x a g <trigger>
;; <after> C-x a g <trigger>
;; <before> C-x a i g <trigger>
;;
;; M-x list-abbrevs
;; M-x edit-abbrevs
;;   file
;;   ^^^^
;;   "<trigger>" counts "<abbrev>"
;;
;; ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;; Abbrev
;; from the file.
```

```
;;
;; M-Space
;; <trigger> Space ... M-x abbrev-mode
;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(setq abbrev-file-name "~/.emacs.d/abbrev/.abbrev-defs"
      save-abbrevs t
)
(quietly-read-abbrev-file)
(define-key esc-map " " 'expand-abbrev)
;;; For Japanese
(add-hook 'pre-command-hook (lambda ()
                              (setq abbrev-mode nil)
                              )
)


;; ,.,.,.,...,.,.,..,.,.,.,.,.,.,.,.,.,
;; Dabbrev
;; from buffers.
;; o English
;; x Japanese
;;
;; <trigger> M-/      ... dabbrev-expand
;;   Prev: M-/
;;   Next: C-u M-/
;; M-C-/              ... dabbrev-completion
;; C-^
;(global-set-key "\C-^" 'dabbrev-expand)
(global-set-key "\C-^" 'davvrev-completion)


;; ,.,.,.,.,.,.,.,.,.,.,.,..,.,.,.,.,.,.
;; Completion
;; from buffers.
;;
;; <trigger> C-Return ... expand
;; <trigger> M-Return ... complete
;;   Prev: C-Return
;;   Next: C-u C-Return
(dynamic-completion-mode)


;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ispell (スペル・チェック)
;;
;; * ispell
;; * app-text/ispell
;; * /usr/lib/ispell/english.hash ... ispell
;; * ~/.ispell_hashfile
;; * http://fmg-www.cs.ucla.edu/geoff/ispell.html
;; * sys-apps/miscfiles
;; * /usr/share/dict/words        ... lookup
;; * http://savannah.gnu.org/projects/miscfiles/
;;
;; <region> M-x ispell-region
;;   Yes:  0-9
;;   No:   Space
;;   List: l
;;   Edit: i
;;
;; <region> M-x flyspell-region
;;
;; C-]                            ... completion
;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(eval-after-load "ispell" '(progn
                            (add-to-list 'ispell-skip-region-alist '("[^\000-\377]+"))
                            )
)
```

```
(setq ispell-grep-command "/usr/bin/grep"
;;      ispell-alternate-dictionary "~/.emacs.d/ispell/english.hash"
)
;;(setenv "DICTIONAY" "~/.emacs.d/ispell/english.hash")
(global-set-key "\C-]" 'ispell-complete-word)
```

3.10 *tags (プログラミングに特化した検索機能)

総合開発環境の話まではしてません。少なくとも此れに加えデバッグの支援も必要です。

ソースコードに関数の呼出しが書かれていれば、其の関数の内容を確認するため定義場所へ移動出来ます。また、其れ等識別子を保存した一時ファイルを参考にした補完も可能です。実装には etags (Emacs), ctags (dev-util/ctags), gtags (dev-util/global) 等が有ります。

『dev-util/global ; カーソル下の識別子 (変数や関数) が定義されている場所へ移動する』に tags.sh を載せています。

....[code:3.10 ※]

```
;;; emacs-23
;;; * dev-util/ctags
;;; * dev-util/global
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;; dev-util/ctags
;; カーソル位置の参照が定義している場所への移動。
;; ./tags が必要。
;;
;; * STANDARD LIBRARY
;;
;; M-x visit-tags-table <file>
;; M-x gtags-visit-rootdir <directory>
;; M-x list-tags
;;
;; M-.
;;   prev: M-*
;;   next: C-u M-.
;; M-x tags-search <search>
;;   prev: M-*
;;   next: M-,
;; M-x tags-query-replace <search> <replace>
;;   prev: M-*
;;   next: M-,
;; ;:;::;:;::;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;


;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; app-editors/emacs
;; カーソル位置の参照が定義している場所への移動。
;; ./TAGS が必要。
;;
;; same as ctags.
;;


;; ,..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
```

```
;; dev-util/global
;; カーソル位置の参照が定義している場所への移動。
;; ./GTAGS が必要。
;;
;; M-x visit-tags-table <file>
;; M-x gtags-visit-rootdir <directory>
;;
;; M-x tags-search
;; M-x tags-query-replace
;;
(autoload 'gtags-mode "gtags" nil t)
(setq gtags-mode-hook
      '(lambda ()
         (local-set-key "\C-ct" 'gtags-find-tag)
         (local-set-key "\C-cr" 'gtags-find-rtag)
         (local-set-key "\C-cf" 'gtags-find-symbol)
         (local-set-key "\C-cp" 'gtags-pop-stack)
         )
)
```

3.11 Emacs 独自のスクリプト言語 (Lisp 言語の亜種)

Emacs はキーボード操作を主とします。幾つか組み合わせる、つまり端から見れば黒客の鍵打秘術に見える程の長い動作も有るでしょう。一連のキーボード動作をマクロとしてまとめ、別名を与える事が出来ます。

其れ所か Emacs 独自のスクリプト言語 (Lisp 言語の亜種) を搭載しています。流石にエディタの範疇外かも知れません。

.....[code:3.11 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;;
;; <lisp> C-x C-e
;;
;; M-x <emacs_function>
;; ,.,..,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; Cancel
;;
;; * STANDARD LIBRARY
;;
;; C-g
```

3.12 シェルとの連携

GNU Emacs はウィンドウ内でメーラやブラウザ等外部プログラムを起動する事が出来ます。此れもエディタの範疇外かも知れません。しかしバッファ内容を入力としたり、出力結果を文章へ残す等はエディタの範疇かも知れません。DBus や Xwindow のクリップ・ボード等と連携すれば可能ですが、Emacs だけで同じ事を実現します。‘‘<region> C-[|’’、それから‘‘C-[x eshell’’ もしくは ‘‘C-[x shell’’ でシェル・

モードが起動します。

以下は app-terms/xterm が文字コード UTF-8 でも shell-mode 内は文字コード EUC-JP に成る設定をしています。付録 C に xterm.sh も載せてみます。

.....[code:3.12 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;; SHELL
;;
;; M-x eshell
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; M-x shell
;;   prev history: M-p
;;   next history: M-n
;;   completion: <command> [Tab]...
;;
;; <region> M-| <shell_command>
;;
(add-hook 'comint-output-filter-functions 'comint-watch-for-password-prompt)
(autoload 'ansi-color-for-comint-mode-on "ansi-color" nil t)
(add-hook 'shell-mode-hook '(lambda ()
                              (pcomplete-shell-setup)
                              (ansi-color-for-comint-mode-on)
                              (set-buffer-process-coding-system 'euc-jp-unix 'utf-8)
                              )
)
(setq explicit-shell-file-name "/bin/bash"
      shell-file-name "/bin/bash"
      ;; --debug --debugger --protected -c
      shell-command-option ""
)
```

3.13 着色

表示されるテキストを装飾します。

例えば全角空白やタブや行末空白を半角空白と区別したいなら着色を利用します。同様にカーソル位置の括弧が示す範囲 (対になる括弧との間にある文字列全て) を強調させてみます。

.....[code:3.13 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
(defface my-face-b-1 '((t (:background "green"))) nil)
(defface my-face-b-2 '((t (:background "green"))) nil)
(defface my-face-u-1 '((t (:background "green"))) nil)
(defvar my-face-b-1 'my-face-b-1)
(defvar my-face-b-2 'my-face-b-2)
(defvar my-face-u-1 'my-face-u-1)
(defadvice font-lock-mode (before my-font-lock-mode ())
  (font-lock-add-keywords major-mode
```

```
                          (list
                          '("  " 0 my-face-b-1 append)
                          '("\t" 0 my-face-b-2 append)
                          '("[ \t]+$" 0 my-face-u-1 append)
                          )
  )
)
(ad-enable-advice 'font-lock-mode 'before 'my-font-lock-mode)
(ad-activate 'font-lock-mode)


;;; emacs-23
;;; * STANDARD LIBRARY
(setq blink-matching-parent t
      blink-matching-delay 0.1
      blink-matching-parent-distance 102400
      show-paren-style 'expression
      font-lock-support-mode 'jit-lock-mode
)
(global-font-lock-mode t)
(show-paren-mode t)
(set-face-background 'show-paren-match-face "red")
(set-face-foreground 'show-paren-match-face "black")
```

3.14 Template (テンプレート)

バッファを新規作成する際にモードもしくは拡張子に基づいてテンプレートを導入出来ます。テンプレートの内容は自由です。テンプレートは各自で用意します。

.....[code:3.14 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
(auto-insert-mode t)
(setq auto-insert-query nil
      auto-insert-directory "~/.emacs.d/insert/"
      auto-insert-alist (append
                         '(
                           ;; "Regex" or Mode
                           ;; ISO/IEC  9945:2009 POSIX
                           ;; BASH (Not SH) --- interactive
;;                           (shell-mode . "template.sh")
                           ;; ISO/IEC  9945:2009 POSIX
                           ;; BASH (Not SH) --- script
                           ("\\.sh\\'" . "template.sh")
                           ;; ISO/IEC  9075:2008 SQL
                           ("\\.sql\\'" . "template.sql")
                           (sql-mode . "template.sql")
                           ;; ISO/IEC  9899:1999 C
                           ;; ISO/IEC 14882:1998 C++
                           ;; ISO/IEC 23270:2006 C#
;;                           (cc-mode . "template.c")
                           (c-mode . "template.c")
                           (c++-mode . "template.cpp")
                           ;; ISO/IEC 16262:2002 ECMAScript
                           ;; ISO/IEC 22537:2006 E4X
                           ("\\.js\\'" . "template.js")
                           (javascript-mode . "template.js")
                           ;; ISO 32000:2008 PDF
                           ("\\.tex\\'" . "template.tex")
```

```
                           ;; ISO 15022:2003 XML
                           ("\\.\\(xml\\)\\'" . "template.xml")
                           ("\\(rss.xml\\|\\.rdf\\|\\.rss\\)\\'" . "template.rdf")
                           ("\\.\\(html\\|xhtml\\)\\'" . "template.html")
                           ("\\.css\\'" . "template.css")
                           ("\\.svg\\'" . "template.svg")
                           ("\\.el\\'" . "template.el")
                           ("\\.py\\'" . "template.py")
                           (python-mode . "template.py")
                           ("Makefile\\'" . "template.Makefile")
                           ("Doxyfile\\'" . "template.Doxyfile")
                           ))
)
```

3.15 Undo (取消)

直前の処理を取り消せます。レジスタですが、独立させ明記してみます。

....[code:3.15 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;; ,.,..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; Undo
;;
;; * STANDARD LIBRARY
;;
;; C-x u    ... one
;; C-x C-u  ... all
;;
(global-set-key "\C-x\C-u" 'revert-buffer)
```

3.16 見出し以外を折り畳み、全体像を把握し易くする

Fold は括弧内を閉じさせ、全体を見渡せる様にします。

Outline は予め決めていた見出し行以外を閉じさせ、全体を見渡せる様にします。

....[code:3.16 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; Fold
;; foo {            >>> foo...
;; }
;;
;; C-u <level> C-x $       ... Selective display
;; C-x $
;;
```

```
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; Outline
;; * chapter        >>> * chapter...
;; foo
;; ** section
;; bar
;; *** subsection
;; baz
;;
;; * STANDARD LIBRARY
;;
;; M-x outline-mode
;; M-x outline-minor-mode
;;
;; C-c C-o                 ... toggle-children
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
;;; C-c @ >>> M-
;(setq outline-minor-mode-prefix "\M-")
(global-set-key "\C-c\C-o" 'outline-toggle-children)
;(add-hook 'yahtml-mode-hook
;         '(lambda ()
;         (setq outline-regexp "<\\(h1\\|h2.\\|h3..\\|h4...\\|h5....\\|h6.....\\)")
;         (outline-minor-mode t)
;         )
;)
```

3.17 カーソル位置の下のある単語を引金とする機能

カーソル位置の下にある単語を読み取り、単語を引数として何かを起動させます。

以下は辞書を引く設定 (``~/.emacs.el'' と ``~/.lookup.el'') です。

....[code:3.17 ※]

```
;;; emacs-23
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; FFAP: Find File (or URL) At Point
;; カーソル位置のファイル名で C-x C-f
;;
;; * STANDARD LIBRARY
;;
;; C-x C-d <directory> --- list
;; C-x C-v <file>      --- open
;;
;;; emacs-23
(ffap-bindings)
(setq ffap-c-path (list
                  "/usr/include/"
                  "~/include/"
                  )
      ffap-kpathsea-depth 2
      ffap-rfc-path "http://www.minokasago.org/labo/RFC/rfc%s-jp.html"
;;      ffap-rfc-path "http://www.ring.gr.jp/archives/doc/RFC/rfc%s.txt"
)
(require 'url-vars)


;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
```

```
;; SKK + Kakasi
;; 範囲内を「振り仮名」化
;;
;; * app-i18n/kakasi
;;
;; <region> C-u M-x skk-gyakubiki-message
;; <region> M-x skk-hurigana-region
;;


;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Lookup (辞書検索コマンド)
;; - See .lookup.el
;;
;; * dev-libs/eb
;; * app-dicts/eblook
;;
;; * app-emacs/lookup
;;
;; * FreePWING Dictionaries
;; * http://openlab.jp/edict/fpw/
;;
;; word         完全一致
;; *word        後方一致
;; word*        前方一致
;; =keyword     条件検索
;;
;; C-c C-p <search>
;; <region> C-c C-r
;;   Up:   C-h
;;   Down: C-m
;;   Prev: p
;;   Next: n
;; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(autoload 'lookup "lookup" nil t)
(autoload 'lookup-region "lookup" nil t)
(autoload 'lookup-pattern "lookup" nil t)
(global-set-key "\C-c\C-l" 'lookup)
(global-set-key "\C-c\C-r" 'lookup-region)
(global-set-key "\C-c\C-p" 'lookup-pattern)


;;
;; ~/.lookup.el
;;
(setq ndeb-process-coding-system 'euc-jp
      lookup-process-coding-system 'euc-jp
      lookup-search-agents
      '(
      ;; CATALOG
      ;;
      ;; 漢字読み引き
      (ndeb "~/.emacs.d/dictionary/FOKS")
      ;; 百科事典
      (ndeb "~/.emacs.d/dictionary/PDD_Hyakka")
      ;; 人名辞典
      (ndeb "~/.emacs.d/dictionary/PDD_Jinmei")
      ;; 地理辞書 (英)
      (ndeb "~/.emacs.d/dictionary/WFB")
      ;; 歴史辞書
      (ndeb "~/.emacs.d/dictionary/MHDB")
      ;; 英英
      (ndeb "~/.emacs.d/dictionary/WONDERNET")
      ;; 英英
      (ndeb "~/.emacs.d/dictionary/WEB")
      ;; 和独
      (ndeb "~/.emacs.d/dictionary/WADOKU")
      ;; 和英
```

```
      (ndeb "~/.emacs.d/dictionary/EDICT")
      ;; 英和
      (ndeb "~/.emacs.d/dictionary/GENE95")
      ;; 不明解
      (ndeb "~/.emacs.d/dictionary/FUMEIKAI")
      ;; 略語
      (ndeb "~/.emacs.d/dictionary/RYAKU")
      ;; 花
      (ndeb "~/.emacs.d/dictionary/FLOWER")
      ;; 生理化学
      (ndeb "~/.emacs.d/dictionary/LSD")
      ;; 通信用語
      ;; (ndeb "~/.emacs.d/dictionary/WDIC")
      ;; Hacher
      ;; (ndeb "~/.emacs.d/dictionary/JARGON")
      ;; 聖人
      ;; (ndeb "~/.emacs.d/dictionary/BIBLENAM")
      ;; 悪魔
      ;; (ndeb "~/.emacs.d/dictionary/DEVIL")
      ;; 聖書
      ;; (ndeb "~/.emacs.d/dictionary/EBD")
      ;; CG 用語
      ;; (ndeb "~/.emacs.d/dictionary/CG")
      ;;
      (ndkks)
      ;; * sys-apps/miscfiles
      ;; * /usr/share/dict/words        ... lookup
      ;; * http://savannah.gnu.org/projects/miscfiles/
      (ndspell)
      )
      ndspell-words-dictionary "/usr/share/dict/words"
      lookup-use-kakasi t
      lookup-use-bitmap t
)
```

3.18 Save (自動保存)

事故への備えです。

以下の設定は外部プログラムを用いてヴァージョン管理をしている環境を前提にしていますので注意して下さい。

....[code:3.18 ※]

```
;; ;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
;; 自動保存
;;
;; * STANDARD LIBRARY
;;
;; ALWAYS
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; #foo.bar#
;;
;; M-x recover-file
;; M-x recover-session
;;;
(auto-save-mode t)
```

```
(setq auto-save-default t
      auto-save-list-file-prefix "~/.emacs.d/recover/"
      delete-auto-save-files t
)

;; .,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; foo.bar~
;;
;; AUTO
;;;
(setq make-backup-files t
      backup-by-copying t
      backup-by-copying-when-linked t
      backup-by-copying-when-mismatch t
)
(defun make-backup-file-name (filename)
  (expand-file-name
    (concat (file-name-nondirectory filename) "~")
            "~/.emacs.d/backup/")
)
```

3.19 日本語出力

Emacs は 1983 年から始まりますが、多言語化は 1987 年から始まります。ISO-2022 形式、Mule 形式、UTF-8 形式と成長します。ところが Emacs が認識に成功していても外部プログラムのパス・データベースやフォント・エンジンが失敗すると日本語が映りません。外部プログラムのほとんどが Xwindow 向けのものですが、Framebuffer 向けのものでも日本語は表示出来ます。ISO 10646 から異体字まで、標準化の明るい話と暗い話が長くなるので此処での解説を避けます。書籍には詳しく書かれているはずです。

以下は x11-terms/xterm (UTF-8) 端末内で失敗した場合に調整してみて下さい。それでも駄目なら外部プログラムが原因かも知れません。付録 C に xterm.sh も載せてみます。

.....[code:3.19 ※]

```
;;; emacs-23
;;; * STANDARD LIBRARY
;(set-terminal-coding-system 'utf-8)
;(set-keyboard-coding-system 'utf-8)
;;;(set-selection-coding-system 'euc-jp)
;;;(set-buffer-process-coding-system 'euc-jp-unix 'iso-8859-1)
;(setq file-name-coding-system 'utf-8)
;(setq default-buffer-file-coding-system 'utf-8)
(set-language-environment 'Japanese)

;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; Encoding and Setting
;;
;; * STANDARD LIBRARY
;;
;; C-x C-m f [utf-8/sjis/eucjp]...
;; C-x C-m k [utf-8/sjis/eucjp]...
;; C-x C-m t [utf-8/sjis/eucjp]...
;; C-x C-m p [utf-8/sjis/eucjp]...
;; C-x C-m x [utf-8/sjis/eucjp]...
;; C-x C-m c [utf-8/sjis/eucjp]...
```

```
;;
```

3.20 日本語入力

‘‘C-¥’’ で日本語入力が可能になります。

しかし外部日本語入力プログラムを経由する様に拡張しているのが大半です。メールでもブラウザでも同じ外部日本語入力プログラムを経由すれば、日本語入力に対して統一された操作になるからです。外部プログラムには何種類か有り、好みが分れます。書籍には詳しく書かれているはずです。

以下は拡張パッケージなのに ELisp だけで実装された、つまり Emacs が動くなら外部プログラムがマージされていなくても構わない、そんな珍しい日本語入力方法の設定 (~/.emacs.el と ~/.skk.el) です。例え日本語が表示されない (Xwindow や Framebuffer がマージされていない) 目隠し状況でも日本語は確かに記述されます。

.....[code:3.20 ※]

```
;;; emacs-23
;;; * app-emacs/ddskk
;;
;; C-c C-j           ... Japanese
;;   hiragana: C-j
;;   katakana: q
;;   roma:     l
;;   zenkaku:  L
;;
;; <KaKiKuKeKo> Space
;; / <abbrev> Space
;; / <ispell> Tab
;; Q <region> Space
;;
;; \
;; @
;; $
(require 'skk-autoloads)
(global-set-key "\C-c\C-j" 'skk-mode)
(setq skk-show-annotation t
      ;; C-o
      skk-annotation-show-wikipedia-url t
)


;;
;; ~/.skk.el
;;
(require 'skk-study)
(setq skk-jisyo "~/.skk-jisyo"
      skk-backup-jisyo "~/.skk-jisyo.BAK"
      skk-server-host "localhost"
      skk-server-portnum "1178"
      skk-search-prog-list (append
                            '(
                            ;; 国語辞書
                            ;; (skk-search-jisyo-file "~/.emacs.d/dictionary/SKK/SKK-JISYO.L" 10000 t)
```

```
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.L" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.JIS2" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.JIS2004" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.JIS3_4" 10000 t)
                        ;; 異体字
                        ;; (skk-search-jisyo-file "~/.emacs.d/dictionary/SKK/SKK-JISYO.itaiji" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.itaiji" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.itaiji.JIS3_4" 10000 t)
                        ;; 法律用語
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.law" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.propernoun" 10000 t)
                        ;; 英和
                        ;; (skk-search-jisyo-file "~/.emacs.d/dictionary/SKK/SKK-JISYO.edict" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.edict" 10000 t)
                        ;; 記号辞典
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.assoc" 10000 t)
                        ;; 漢字が既に混ざった状態から変換
                        ;; (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.mazegaki" 10000 t)
                        ;; app-i18n/kakasi
                        (skk-okuri-search)
                        ;; 住所
                        ;; (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.geo" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.geo" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.station" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.zipcode" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.office.zipcode" 10000 t)
                        ;; 人名
                        ;; (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.jinmei" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.jinmei" 10000 t)
                        (skk-search-jisyo-file "/usr/share/skk/SKK-JISYO.fullname" 10000 t)
                        ;; app-dicts/eblookup
                        ;;(skk-lookup-search)
                        )
                        skk-search-prog-list
                        )
skk-auto-start-henkan t
skk-use-look t
skk-look-recursive-search t
skk-look-use-ispell t
skk-use-numeric-conversion t
skk-egg-like-newline t
;;
skk-auto-insert-parent nil
skk-dcomp-active t
skk-auto-okuri-process t
skk-henkan-strict-okuri-precedence t
;;
skk-echo t
skk-isearch-start-mode 'hiragana
;;
skk-latin-mode-string "A"
skk-jisx0208-latin-mode-string "Ａ"
skk-hiragana-mode-string "あ"
skk-katakana-mode-string "ア"
;;
skk-use-face t
skk-henkan-face 'region
;skk-cursor-default-color "#770000"
;skk-cursor-latin-color "#770000"
;skk-cursor-jisx0208-latin-color "#cc7777"
;skk-cursor-hiragana-color "#7777cc"
;skk-cursor-katakana-color "#77cc77"
skk-cursor-default-color "red"
skk-cursor-latin-color "red"
skk-cursor-jisx0208-latin-color "magenta"
skk-cursor-hiragana-color "cyan"
skk-cursor-katakana-color "blue"
```

)

4 Many people have seen FINE Is Not Emacs, but the one that has its character is THief Isn't Even Fine.

定期的に作文を書いたり、毎日の日記を付けたり、ちょっとメモへ残したり。効果の程は人夫々の様ですが未体験なら一度試してみましょう。私はメモと相性が良かったです。冗談も書いたし、概念も書いたし、とっても満足したのですが、此処では共有するかも知れないメモについて少し個人的感想も書いてみます。オフィス・スイートには触れてません。

メモですから忘れてしまう前に書き始められるものが良いと思っています。落書きを書く前からメタ情報を考えたら駄目で、溜まったメモを其の少ない自動付加されたメタ情報を元に整理してくれるツールが便利でした。メモは意外と奥が深く (題名は要らないけど日付は必要など)、「完成状態」をで無く「仕組み」から考える必要が有る様ですね。

文章を計算機上で (オフィスを越えて) 共有したいのなら ISO 15022:2003 XML とか ISO 32000:2008 PDF とか RFC 2822 message 辺りを意識するのが無難みたいです。www-client/mozilla-firefox なら XML を XHTML へしなくとも XML に直接 CSS が適応出来ます。各々専用の作成ツールが有り、エディタの出番は無いかの様に思えます。

XML で記述して XSL で Web Server Negotiation 計画とか考えた頃もあったのですが、私には合わなかった様です。SVG も断念しました。メール本文へそのまま使っても違和感が無いぐらいの軽い文法 (本当に自由な書式だと XML や PDF へ変換し難いどころかメールとしても読み難い) で文章を残せば XML や PDF へ変換してくれる、そんな我侭を許してくれたのが RST: ReStructuredText 文法や Textile 文法や Muse 文法などでした。愛用してます。エディタで十分でした。

GentooJP さんで文章を書く様になってからは RST の素朴な XML や PDF から広報っぽく少し派手にしようと思いまして、「RST のメモを参考に直接 XHTML を記述して net-print/cups-pdf で PDF を作成」もしくは「RST のメモを参考に直接 TeX を記述して PDF から XML を作成」をしてます。其れなら最初の専用ツールへ話が戻るのですが、何故か専用ツールと相性が悪い (Emacs から離れられない) のです。それから一度決めたキー・バインディングを変えられない苦しさに似たもので、XML や PDF のデザインも一度決めてしまうと根本からの変更はし難いですね。

....[code:4.0 ※]

```
;;; emacs-23
;;; * app-emacs/rst + dev-python/docutils
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;;
;; C-c C-a            ... title/chapter/section
;; C-c C-t            ... list
;; C-c C-u            ... update
;; C-c C-p            ... prev
```

```
;; C-c C-n            ... next
;; <region> C-c C-c   ... .. comment
;; <region> C-c C-e   ... 1. item
;; <region> C-c C-b   ... - item
;; <region> C-c C-r   ...   item
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;;
;; ############################################################
;; ISO 15022:2003 XML? ISO 32000:2008 PDF? RFC 2822 message?
;; ############################################################
;;
;; 定期的に作文を書いたり、毎日の日記を付けたり、ちょっとメモへ残したり。効果の程は人夫々...
;;
;; メモですから忘れてしまう前に書き始められるものが良いと思っています。落書を書く前からメ...
;;
;; 同時に文章を計算機上で（オフィスを越えて）共有したいのなら ISO 15022:2003 XML [1]_ とか...
;;
;; XML で記述して **XSL で Web Server Negotiation 計画** とか考えた頃もあったのですが、私...
;;
;; GentooJP さんで文章を書く様になってからは RST の素朴な XML や PDF から広報っぽく少し派...
;;
;; .. [1] http://xxx
;;
;; ,.,.,.,....,.,.,.,..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;;
;; ############        ... <h1>title</h1>
;; title
;; ############
;;
;; chapter
;; ========== or ::::::::::
;;
;; section
;; ^^^^^^^^^^ or ~~~~~~~~~~
;;
;; subsection          ... <h4>subsection</h4>
;; """""""""" or ''''''''''
;;
;; * bulleted-list     ... <ul><li>bulleted-list</li></ul>
;; * bulleted-list
;; * bulleted-list
;;
;; a. enumerated-list ... <ol><li>enumerated-list</li></ol>
;; b. enumerated-list
;; c. enumerated-list
;;
;; a. * `Section`_
;;    * `Section`_
;;    * `Section`_
;;    * * `SubSection`_
;;      * `SubSection`_
;;      * `SubSection`_
;;
;; paragraph           ... <p>paragraph</p>
;;
;; paragraph::         ... <p>paragraph:</p>
;;
;;   preformatted-text    <pre>preformatted-text</pre>
;;   preformatted-text
;;   preformatted-text
;;
;; > blockquote        ... <blockquote>blockquote</blockquote>
;;
;; definition          ... <dl><dt>definition></dt>
;;   description              <dd>description</dd></dl>
;;
;; mmmmm *em* wwwww    ... <em>em</em>
;; mmmmm **strong** wwwww
```

```
;; mmmmm \< wwwww     ... &lt;
;; .. comments        ... <!-- comments -->
;;
;; mmmmm [number]_ wwwww  <a href="#id">number</a>
;;
;; .. [number] footnote
;;
;; mmmmm [string]_ wwwww  <a href="#id">string</a>
;;
;; .. [string] citation
;;
;; mmmmm link_ wwwww      <a href="http://xxx">link</a>
;;
;; .. _link: http://xxx
;;
;; mmmmm |link| wwwww     <img src="http://xxx" />
;;
;; .. |link| image:: http://xxx
;;
;; +------+------+------+ <table>
;; | text | text | text |
;; +======+======+======+
;; | text | text | text |
;; +======+======+======+
;; | text | text | text |
;; +------+------+------+
;;
(setq default-major-mode 'rst-mode)
(require 'rst)
(add-hook 'rst-adjust-hook 'rst-toc-update)


;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; * YaTeX
;; * app-emacs/yatex
;; * http://www.yatex.org/
;;
;; C-c '
;; C-c g
;;
;; C-c d
;;
;; M-x YaTeX-section-overview
;; C-p/n
;; C-v/M-v
;;
;; C-c > --- comment
;; C-c <
;;
;; C-c i --- indent
;; M-q
;;
;; C-u Tab       --- completion
;; C-c Space
;; C-c a <sign>
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; YaTeX
;;
;; C-c t s --- xdvi -remote
;; C-c t l --- lpr
;; C-c t j --- platex
;; C-c t r --- platex
;; C-c t i --- makeindex
;; C-c t b --- jbibtex
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; YaHTML
;;
;; C-c b Space <tagname>
```

```
;; <region> C-c B Space <tagname>
;; C-c a
;; C-c t p
;; M-Return
;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; Math
;;
;; C-c w
;; : <sign>
;; ; <sign>
;;
;; (autoload 'yatex-mode "yatex" nil t)
;; (autoload 'yahtml-mode "yahtml" nil t)
;; (setq tex-command "platex --src-specials"
;;       dvi2-command (if (eq window-system 'x)
;;                        "pxdvi -expert -p 300 -s 3 -geo 750x575+0+0"
;;                      "dviout")
;;       makeindex-command "mendex"
;;       dviprint-command-format "pdvips %s | lpr"
;;       dviprint-from-format "-p %b"
;;       dviprint-to-format "-l %e"
;;       YaTeX-template-file "~/.emacs.d/insert/template.tex"
;;       YaTex-create-file-prefix-g t
;; ;;      YaTeX-kanji-code 0 ... =
;; ;;      YaTeX-kanji-code 1 ... S
;; ;;      YaTeX-kanji-code 2 ... J
;; ;;      YaTeX-kanji-code 3 ... E
;; ;;      YaTeX-kanji-code 4 ... u
;;       YaTeX-kanji-code 4
;;       YaTeX-use-LaTeX2e t
;;       YaTeX-use-AMS-LaTeX t
;; ;;      YaTeX-latex-message-code 'euc-jp
;;       YaTeX-latex-message-code 'utf-8
;;       YaTeX-no-begend-shortcut t
;;       ;;
;;       yahtml-www-browser "firefox.sh"
;;       yahtml-p-prefered-env-regexp "^\\(body\\|div\\|blockquote\\)"
;;       yahtml-path-url-alist '(
;;                              ("" . "")
;;                              ("" . "")
;;                              ("" . "")
;;                              ("" . "")
;;                              ("" . "")
;;                              ("" . "")
;;                              )
;;       yahtml-form-table '(
;;                           ("div")
;;                           ("p")
;;                           ("blockquote")
;;                           ("dl")
;;                           ("ol")
;;                           ("ul")
;;                           ("table")
;;                           ("form")
;;                           ("a")
;;                           ("img")
;;                           ("em")
;;                           ("strong")
;;                           ("ins")
;;                           ("del")
;;                           )
;; )
;; ;;; For ddskk
;; (add-hook 'skk-mode-hook
;;           (lambda ()
;;             (if (eq major-mode 'yatex-mode)
;;                 (progn
;;                   (define-key skk-j-mode-map "\\" 'self-insert-command)
```

```
;;                   (define-key skk-j-mode-ma- "$" 'YaTeX-insert-dollor)
;;                 )
;;               )
;;             )
;; )
;; ;;;
;; (add-hook 'yatex-mode-load-hook
;;           (function
;;            (lambda ()
;;              (auto-fill-mode -1)
;;              (YaTeX-define-key "w" 'sdic-describe-word)
;;              (YaTeX-define-key "\C-w" 'YaTeX-switch-mode-menu)
;;            )
;;           )
;; )
;; (add-hook 'yatex-mode-hook
;;           (function
;;            (lambda ()
;;              (auto-fill-mode -1)
;;            )
;;           )
;; )
;; ;;; For w3m
;; (autoload 'w3m-goto-url "w3m" nil t)
;; (defadvice yahtml-browse-html
;;   (around w3m-yahtml-browse-html activate compile)
;;   (w3m-goto-url (ad-get-arg 0) t)
;; )
;; ;; For math
;; (setq YaTeX-math-sign-alist-private
;;       '(
;;       ("q" "Q" "(Q)")
;;       ("z" "Z" "ZZ")
;;       ("t" "text" "text")
;;       ("qu" "quad" "__")
;;       ("qq" "qquad" "____")
;;       ("ls" "varlimsup" "___\nlim")
;;       ("li" "varlimsup" "lim\n---")
;;       ("il" "varinjlim" "lim\n-->")
;;       ("pl" "varprojlim" "lim\n<--")
;;       ("st" "text{ s.t. }" "s.t.")
;;       ("bigop" "bigoplus" "_\n(+)~")
;;       ("bigot" "bigotimes" "_\n(x)\n ~")
;;       )
;; )


;; ,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
;; * howm
;; * app-emacs/howm
;; * http://howm.sourceforge.jp/index-j.html
;;
;; C-c , ,
;;
;; (autoload 'howm-menu "howm" nil t)
;; (global-set-key "\C-c,," 'howm-menu)
;; (eval-after-load "howm-mode" '(progn
;;                                 (define-key howm-mode-map [tab] 'action-lock-goto-next-link)
;;                                 (define-key howm-mode-map [(meta tab)] 'action-lock-goto-previous-link)
;;                               )
;; )
;; (mapc (lambda (f)
;;       (autoload f "" nil t))
;;       '(howm-menu howm-list-all
;;                   howm-list-recent
;;                   howm-list-grep
;;                   howm-create
;;                   howm-keyword-to-kill-ring)
```

```
;; )
;; (add-hook 'howm-mode-hook '(lambda ()
;;                              (progn
;;                                (defun howm-create-file-with-title (title &optional  which-template not-use-f
;;                                  (let ((f (if w3m-current-url
;;                                               (progn
;;                                                 (setq title (if w3m-current-title
;;                                                                 w3m-current-title
;;                                                               ""))
;;                                                 w3m-current-url)
;;                                             buffer-file-name)
;;                                         ))
;;                                    (when (not here)
;;                                      (howm-create-file)
;;                                    )
;;                                    (goto-char (point-max))
;;                                    (when (not (= (point) (point-min)))
;;                                      (delete-blank-lines)
;;                                      (when (not (= (line-beginning-position) (point)))
;;                                        (insert "\n"))
;;                                      (insert "\n")
;;                                    )
;;                                    (let ((p (point))
;;                                          (insert-f (lambda (switch)
;;                                                      (howm-insert-template (if switch title "")
;;                                                                            f which-template (not switch)
;;                                                      )
;;                                                    )
;;                                          )
;;                                          (use-file (not not-use-file))
;;                                         )
;;                                      (funcall insert-f not-use-file)
;;                                      (undo-boundary)
;;                                      (delete-region p (point-max))
;;                                      (funcall insert-f use-file)
;;                                    )
;;                                    (when (and (stringp f)
;;                                               (string-match "^http://" f))
;;                                      (kill-line 2)
;;                                      (insert "\n\n" f)
;;                                      (forward-line -2)
;;                                    )
;;                                    (howm-set-mode)
;;                                  )
;;                                )
;;                                (setq howm-directory "~/.emacs.d/howm/"
;;                                      howm-file-name-format "%Y/%m/%Y-%m-%d-%H%M%S.howm"
;;                                      howm-keyword-file "~/.emacs.d/howm/.howm-keys"
;;                                      howm-menu-file "0000-00-00-000000.howm"
;;                                      howm-view-grep-parse-line "^\\(\\(\\(\a-zA-Z]:/\\)?[^:]*\\.howm\\):\\(\0-9
;;                                      howm-excluded-file-regexp "/\\.#\\|[~#]$\\|\\.bak$\\|/images/\\|/\\.git
;;                                      howm-menu-top nil
;;                                      hown-menu-lang 'ja
;;                                      howm-list-recent-title t
;;                                      howm-list-all-title t
;;                                      howm-view-summary-persistent nil
;;                                      howm-menu-schedule-days-before 32
;;                                      howm-menu-schedule-days 8
;;                                      howm-view-search-in-result-correctly t
;; ;;                                   howm-menu-recent-num 20
;; ;;                                   howm-view-summary-window-size 10
;;                                      ;;howm-search-other-dir t
;;                                      howm-search-path '(
;;                                                         "~/Mail/"
;;                                                         "~/News/")
;; ;;                                   howm-view-use-grep nil
;;                                      howm-menu-refresh-after-save t
;;                                      howm-refresh-after-save t
```

```
;;                                  )
;;                                )
;;                              )
;; )


;; ,.,..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
;; ChangeLog
;;
;; * STANDARD LIBRARY
;;
;; M-x add-change-log-entry
;; M-x add-change-log-entry-other-window
;; M-x change-log-merge
;;
(setq change-log-default-name "ChangeLog"
      add-log-full-name "Blurgle Blurgle"
      add-log-mailing-address "blurgle.mailAAAAAAAAAAgmail.com"
)
;;; M-x howm-changelog
(defun howm-changelog ()
  (interactive)
  (add-change-log-entry nil (expand-file-name "~/.emacs.d/howm/ChangeLog"))
)
```

5 パッケージ化

ソフトウェアの ebuilds を作成する理由には、構築作業の自動化、検査作業の自動化、パッケージ情報のデータベース化があります。

5.1 「名前を検索」「配布元を検索」「説明文を検索」等

以下は『app-portage/eix ; パッケージ検索』より「GPMS と app-portage/eix」を再掲載してます。

名前検索してみます。

..... [code:5.1a ※] [input:5.1a] [output:5.1a]

```
n/a
```

..... [code:5.1a] [input:5.1a ※] [output:5.1a]

```
<< software.sh name virtual
```

..... [code:5.1a] [input:5.1a] [output:5.1a ※]

```
<< eix --name --cache-file /var/cache/eix --verbose --regex virtual

* app-emulation/virtualbox
     Available versions:  4.0.12 (~)4.1.4 (~)4.1.6-r1 **9999 {+additions alsa doc elibc_FreeBSD extensions headless java ker
     Best versions/slot:  (~)4.1.6-r1
     Homepage:            http://www.virtualbox.org/
     Description:         Softwarefamily of powerful x86 virtualization
     License:             GPL-2

...snip...

Found 12 matches.
```

配布元で検索してみます。

..... [code:5.1b ※] [input:5.1b] [output:5.1b]

```
n/a
```

..... [code:5.1b] [input:5.1b ※] [output:5.1b]

```
<< software.sh uri virtual
```

..... [code:5.1b] [input:5.1b] [output:5.1b ※]

```
<< eix --homepage --cache-file /var/cache/eix --verbose --regex virtual

* app-backup/vzdump
     Available versions:  (~)1.1 (~)1.2
     Best versions/slot:  (~)1.2
     Homepage:            http://www.proxmox.com/cms_proxmox/en/virtualization/openvz/vzdump/
     Description:         A utility to make consistent snapshots of running OpenVZ containers.
     License:             GPL-2

...snip...

Found 15 matches.
```

説明文で検索してみます。

.....[code:5.1c ※] [input:5.1c] [output:5.1c]

```
n/a
```

.....[code:5.1c] [input:5.1c ※] [output:5.1c]

```
<< software.sh description virtual
```

.....[code:5.1c] [input:5.1c] [output:5.1c ※]

```
<< eix --description --cache-file /var/cache/eix --verbose --regex virtual

* app-cdr/burn-cd
     Available versions:  1.7.4 (~)1.8.0
     Best versions/slot:  (~)1.8.0
     Homepage:            http://burn-cd.sourceforge.net
     Description:         Smart console frontend for virtual/cdrtools and dvd+rw-tools
     License:             LGPL-2

...snip...

Found 248 matches.
```

5.2 「あのパッケージはどんなコマンドを提供しているのか」「其の USEflag の意味は何か」「此のファイルは何のパッケージに含まれるのか」等

以下は『app-portage/portage-utils ; ユーザ向け ebuild 資料閲覧用具』より「GPMS と app-portage/portage-utils」を再掲載してます。

パッケージに含まれたファイルを確認します。

.....[code:5.2a ※] [input:5.2a] [output:5.2a]

```
n/a
```

. . . . . [code:5.2a] [input:5.2a ※] [output:5.2a]

```
<< package.sh list cfg-update
```

. . . . . [code:5.2a] [input:5.2a] [output:5.2a ※]

```
<< qlist --verbose --exact cfg-update

/etc/cfg-update.conf
/etc/cfg-update.hosts
/usr/bin/emerge_with_indexing_for_cfg-update_phphelper
/usr/bin/cfg-update_phphelper
/usr/bin/emerge_with_indexing_for_cfg-update_bashhelper
/usr/bin/emerge_with_indexing_for_cfg-update
/usr/bin/cfg-update
/usr/share/man/man8/cfg-update.8.bz2
/usr/share/doc/cfg-update-1.8.7/ChangeLog.bz2
/usr/lib/cfg-update/.bashrc
/usr/lib/cfg-update/cfg-update
/usr/lib/cfg-update/test.tgz
/usr/lib/cfg-update/cfg-update_indexing
/var/lib/cfg-update/.keep_app-portage_cfg-update-0
```

USEflag の意味を確認します。

. . . . . [code:5.2b ※] [input:5.2b] [output:5.2b]

```
n/a
```

. . . . . [code:5.2b] [input:5.2b ※] [output:5.2b]

```
<< package.sh description postgres
```

. . . . . [code:5.2b] [input:5.2b] [output:5.2b ※]

```
<< quse --verbose --describe postgres

 global:postgres: Adds support for the postgresql database
 local:postgres:app-admin/ulogd: Build PGSQL output plugin to save packets in a postgres database.
 local:postgres:dev-php/PEAR-HTTP_Download: Send postgres LOBs without buffering
 local:postgres:net-misc/cfengine: Use dev-db/postgresql-server as database backend, default would be Berkeley DB (sys-libs/
```

対象ファイルがどのパッケージによりマージされたのかを確認します。

. . . . . [code:5.2c ※] [input:5.2c] [output:5.2c]

```
n/a
```

. . . . . [code:5.2c] [input:5.2c ※] [output:5.2c]

```
<< package.sh owner passwd
```

. . . . . [code:5.2c] [input:5.2c] [output:5.2c ※]

```
<< qfile --verbose --root-prefix --slots passwd

sys-apps/shadow-4.1.4.3:0 (/bin/passwd)
sys-apps/shadow-4.1.4.3:0 (/etc/pam.d/passwd)
sys-apps/shadow-4.1.4.3:0 (/usr/bin/passwd)
sys-apps/baselayout-2.1:0 (/usr/share/baselayout/passwd)
```

5.3　構築作業の自動化

『GPMS (0) — メタ・ディストリビューションの布石 (EAPI)』等を参考にして下さい。

5.4　検査作業の自動化

『C/C++ の井戸端会議 (0x05) — Gesellschaft ⇔ Gemeinschaft』等を参考にして下さい。

付録 A　ChangeLog

```
[2011-11-11] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* version 0.3.0
  「パッケージ化」を追加。
[2011-01-03] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* version 0.2.1
  「Many people have seen FINE Is Not Emacs, but the one that has its character is THief Isn't Even Fine.」
へ追記。
[2010-05-08] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* version 0.2.0
  media-gfx/gimp での体験から、
  構成を見直しほぼ全面的に書き直す。
  視座を教えてくれたのはレイヤ機能でした。
[2009-02-07] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* version 0.1.0
  各ページのファイル・ネームは無差別メモ貯蔵庫から Gentoo 記事へまとめた初作成日（の目安）です。
  ウェブサイトの更新履歴（ChangeLog.pdf）は初公開日が基準になります。
  RDF の資源帳簿（index.rdf）は最終更新日も含まれます。
  ヴァージョンについては、major.minor.patch です。
  major は互換性を失う程の変更回数、
  minor は目次が増減する程の変更回数、
  patch は誤字脱字などの変更回数です（但し私が手動で変更した数えられる限りにおいてのみ)。
  鮮度は初公開日が基準ですが、
  具体的な数値などを記述している記事は少ないので、読める事は読めると思われます。
* 手順は Gentoo 本家にもありますし、書籍でも調べられます。
```

付録 B　emacs.sh

本来ならシェル・スクリプトは ISO/IEC 9945 POSIX に準拠させるのが妥当なのですが、Gentoo の ebuild が独自拡張の Bash 言語を採用しているため、Gentoo が動いている環境なら Bash 言語に対応するはずなので Bash シェル・スクリプトです。共通ライブラリは『私の贅沢 (子 し) — 此のウェブサイトに登場する Bash シェル・スクリプトに必要な共通ライブラリ』に有ります。因みに Gentoo の ebuild をどの GPM (portage, paludis, pkgcore, entropy, ...) にも対応させたいのなら規格 EAPI に準拠させる、具体的には app-doc/pms と app-portage/eclass-manpages と app-doc/devmanual を参考に記述してみて下さい。

時間的制約があるので、設計も検査もせず動いた所で完成としてます。此の様なスクラッチと呼ばれる書捨て (←一巡目の雰囲気を掴むためだけのプロトタイプの事、因みに二巡目はゼロから書き直して三巡目以降は二巡目を改良する形で開発する) は初学者目線に近く面白いシリーズと思っているのですが、ISO/IEC 14882 C++ に Boost で書いた方が面白いシリーズになったのではないか、とは今でも思います。

..... [code:102.1 ※] [input:102.1] [output:102.1]

```
#!/bin/bash
# -*- coding: utf-8 -*-
#
# [2011-10-31] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.6.0
#     library_message.sh, library_string.sh, library_math.sh, library_datetime.sh, library_path.sh, library_trap.sh, library
# [2011-05-20] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.5.0
#     app-shells/bash-3.2
#     BSD-3 (modified BSD license)
# [2011-02-15] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.4.0
#     app-shells/bash-4.1
#     BSD-2 (FreeBSD license)
# [2009-11-01] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.3.0
#     app-shells/bash-4.0
#     BSD-3 (modified BSD license)
# [2005-08-13] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.2.0
#     app-shells/bash-2.95
#     BSD-4 (original BSD license)
# [2002-02-01] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
#  * version-0.1.0
#     initialization
#
#
# Copyright (c) 2002--2013, Blurgle Blurgle
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# 1. Redistributions of source code must retain the above copyright notice,
#    this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in the
#    documentation and/or other materials provided with the distribution.
# 3. Neither the name of the <ORGANIZATION> nor the names of its
```

```
#    contributors may be used to endorse or promote products derived from
#    this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
# ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES: LOSS OF USE, DATA, OR PROFITS: OR BUSINESS
# INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
# CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
# POSSIBILITY OF SUCH DAMAGE.
#



# Location Hint
# See type -a <name>
#    a) /bin/
#    b) /sbin/
#    c) /usr/bin/
#    d) /usr/sbin/
#    e) /usr/local/bin/
#    f) command_name  ... command command_name
#    g) builtin_name  ... builtin builtin_name
#    h) alias_name
#    i) function_name
# o  B=""             ... use ${PATH}
# x  B="/bin/"
# v  B="/mnt/gentoo/bin/"
# v  B="/bin/busybox "
declare -r F=""
declare -r B=""
declare -r S=""
declare -r UB=""
declare -r US=""
declare -r UL=""


# Boolean
declare -i -r true=0
declare -i -r false=1
declare -i -r error=2


# Bash Library
${F}source library_message.sh
${F}source library_string.sh
${F}source library_math.sh
${F}source library_datetime.sh
${F}source library_path.sh
${F}source library_trap.sh
${F}source library_assertion.sh



# ################################
# Arguments: any type ($1 -- $9)
# Return:    status_code
# ################################
function main() {
    # declare variable="10 11 12 13 1000"
    # echo ${variable:<start>:<length>}
    # echo ${variable/<search>/<replace>}
```

```
    # echo ${variable//<search>/<replace>}
    # echo ${variable#<head>}
    # echo ${variable##<head>}
    # echo ${variable%<tail>}
    # echo ${variable%%<tail>}
    # declare -a array=("")
    # array[0]=10
    # array[1]=11
    # array[2]=12
    # array[3]=13
    # array[100]=1000
    # echo ${array}     ... 10
    # echo ${array[@]}  ... 10 11 12 13 1000
    # echo ${!array[@]} ... 0 1 2 3 100
    # echo ${#array[@]} ... 5
    # echo ${array[${#array[@]}]} ... ERROR
    declare -i number_of_arguments=$((${#@} -1))
    declare -i count=0

    declare action="${F}__gui"
    declare mode="none"
    declare -i status_code=${true}
    declare command="${UB}emacs"
    declare -a array_of_parameter=("")
    declare -i n=0

    declare destination=""
    declare source=""


    # for analyzing, it has three way.
    # a) __parse_commandline_arguments()
    # b) __getopts()
    # c) __shell_getopts()
    # d) __shell_select()
    ${F}__parse_commandline_arguments ${@}
    ${F}eval "${action}"

    ${F}exit ${status_code}
}



# ################################
# Arguments: any type ($1 -- $9)
# Return:    status_code
# ################################
function __parse_commandline_arguments() {
    # <command> --options... <source> <destination>
    #
     if [[ ${mode} == "check" ]] ;
        then
        __set_destination
    fi
    if [[ ${mode} == "check" ]] ;
        then
        __set_source
    fi
    # <command> --options...
    #
    # $BASH_ARGV[@]=(option9 option8 option7...)
    for ((count=0; count <= number_of_arguments; count++))
        do
        case "${BASH_ARGV[count]}" in
            (list|l|show)
                action="${F}__list"
                command="${UL}list.sh"
```

```
                continue
                ;;
            (signal)
                action="${F}__signal"
                command="${UL}signal.sh"
                continue
                ;;
            (21gui|21g|gui21|g21)
                action="${F}__gui"
                command="${UB}emacs-21"
                array_of_parameter[n++]="--user ${USER}"
                array_of_parameter[n++]="--reverse-video"
                array_of_parameter[n++]="--debug-init"
                continue
                ;;
            (21cui|21c|cui21|c21|21)
                action="${F}__cui"
                command="${UB}emacs-21"
                array_of_parameter[n++]="--user ${USER}"
                array_of_parameter[n++]="--no-window"
                array_of_parameter[n++]="--debug-init"
                continue
                ;;
##############################################
            (23gui|23g|gui23|g23)
                action="${F}__gui"
                command="${UB}emacs-23"
                array_of_parameter[n++]="--user ${USER}"
#                array_of_parameter[n++]="--reverse-video"
#                array_of_parameter[n++]="--basic-display"
#                array_of_parameter[n++]="--color never"
#                array_of_parameter[n++]="--color auto"
#                array_of_parameter[n++]="--color always"
#                array_of_parameter[n++]="--color ansi8"
#                array_of_parameter[n++]="--foreground-color "
#                array_of_parameter[n++]="--background-color "
#                array_of_parameter[n++]="--cursor-color "
#                array_of_parameter[n++]="--mouse-color "
#                array_of_parameter[n++]="--font "
#                array_of_parameter[n++]="--border-color "
#                array_of_parameter[n++]="--border-width 0"
#                array_of_parameter[n++]="--internal-border "
#                array_of_parameter[n++]="--line-spacing "
#                array_of_parameter[n++]="--fullheight"
#                array_of_parameter[n++]="--fullwidth"
#                array_of_parameter[n++]="--fullwidth"
#                array_of_parameter[n++]="--geometry "
#                array_of_parameter[n++]="--no-bitmap-icon"
#                array_of_parameter[n++]="--iconic"
#                array_of_parameter[n++]="--no-blinking-cursor"
#                array_of_parameter[n++]="--vertical-scroll-bars"
#                array_of_parameter[n++]="--debug-init"
                continue
                ;;
            (23cui|23c|cui23|c23|23)
                action="${F}__cui"
                command="${UB}emacs-23"
                array_of_parameter[n++]="--user ${USER}"
#                array_of_parameter[n++]="--no-desktop"
                array_of_parameter[n++]="--no-window-system"
                array_of_parameter[n++]="--no-splash"
                array_of_parameter[n++]="--debug-init"
                continue
                ;;
            (23daemon|23d|daemon23|d23|23detach|detach23)
                action="${F}__cui"
                command="${UB}emacs-23"
                array_of_parameter[n++]="--daemon"
```

```
                continue
                ;;
            (23attach|23a|attach23|a23)
                action="${F}__stub"
                command="${UB}emacsclient-emacs-23"
                continue
                ;;
##############################################
            (24gui|24g|gui24|g24)
                action="${F}__gui"
                command="${UB}emacs-24"
                array_of_parameter[n++]="--user ${USER}"
#                array_of_parameter[n++]="--reverse-video"
#                array_of_parameter[n++]="--basic-display"
#                array_of_parameter[n++]="--color never"
#                array_of_parameter[n++]="--color auto"
#                array_of_parameter[n++]="--color always"
#                array_of_parameter[n++]="--color ansi8"
#                array_of_parameter[n++]="--foreground-color "
#                array_of_parameter[n++]="--background-color "
#                array_of_parameter[n++]="--cursor-color "
#                array_of_parameter[n++]="--mouse-color "
#                array_of_parameter[n++]="--font "
#                array_of_parameter[n++]="--border-color "
#                array_of_parameter[n++]="--border-width 0"
#                array_of_parameter[n++]="--internal-border "
#                array_of_parameter[n++]="--line-spacing "
#                array_of_parameter[n++]="--fullheight"
#                array_of_parameter[n++]="--fullwidth"
#                array_of_parameter[n++]="--fullwidth"
#                array_of_parameter[n++]="--geometry "
#                array_of_parameter[n++]="--no-bitmap-icon"
#                array_of_parameter[n++]="--iconic"
#                array_of_parameter[n++]="--no-blinking-cursor"
#                array_of_parameter[n++]="--vertical-scroll-bars"
                continue
                ;;
            (24cui|24c|cui24|c24|24)
                action="${F}__cui"
                command="${UB}emacs-24"
                array_of_parameter[n++]="--user ${USER}"
#                array_of_parameter[n++]="--no-desktop"
                array_of_parameter[n++]="--no-window-system"
                array_of_parameter[n++]="--no-splash"
                continue
                ;;
            (24daemon|24d|daemon24|d24|daemon)
                action="${F}__cui"
                command="${UB}emacs-24"
                array_of_parameter[n++]="--no-site-file"
                array_of_parameter[n++]="--no-init-file"
                array_of_parameter[n++]="--daemon"
                continue
                ;;
            (--help|-h|help)
                action="${F}__help"
                continue
                ;;
            (--version|-v|version)
                action="${F}__version"
                continue
                ;;
            (*)
                source="${BASH_ARGV[count]}"
                source="${source// /?}"
                source="${source//\(/?}"
                source="${source//\)/?}"
                continue
```

```
                ;;
        esac
    done
    return ${true}
}



function __getopts() {
# /usr/bin/getopt --longoptions "longformat" --options "shortformat" -- "$@"
# << script.sh 00 -f=01 --foo=02 -f 03 10 -b=11 --bar=12 -b 13 20 -B=21 --baz=22 -B 23 30
#    ^^^^^^^^^
#    getopt --longoptions "foo,bar:,baz::" --options "fb:B::" -- "$@"
# >> getopt: 無効なオプション -- '='
#    getopt: 無効なオプション -- '0'
#    getopt: 無効なオプション -- '1'
#    getopt: オプション '--foo' は引数を取ることができません
# >> -f -f -b '=11' --bar '12' -b '13' -B '=21' --baz '22' -B '' -- '00' '03' '10' '20' '23' '30'
#    while true
#      do
#      case "$1" in
#        (-f|--foo)
#          # o  no argument
#          # x  argument
#          shift 1
#          ;;
#        (-b|--bar)
#          # x  no argument
#          # o  argument
#          shift 2
#          ;;
#        (-B|--bar)
#          # o  no argument
#          # o  argument
#          case "$2" in
#            ("")
#              # no argument
#              shift 2
#              ;;
#            (*)
#              # argument
#              shift 2
#              ;;
#          esac
#        (--)
#          # a) continue
#          # b) break
#          shift 1
#          break
#          ;;
#        (*)
#          # a) $source
#               $destination
#          # b) exit
#          exit 1
#          ;;
#      esac
#    done
  :
}



function __shell_getopts() {
    # <command> -e
    #           -t
```

```
    #           -c
    #           -u
    # checking short arguments.
    while ${F}getopts "e:t:c:u:" s "${@}"
      do
      case "${s}" in
      (e)
          array_of_parameter[n++]="${OPTARG}"
          continue
          ;;
      (t)
          sWebsiteTitle="${OPTARG}"
          continue
          ;;
      (c)
          sCharset="${OPTARG}"
          continue
          ;;
      (u)
          sURI="${OPTARG}"
          continue
          ;;
      (*)
          continue
          ;;
      esac
    done
    ${F}shift $((${OPTIND} -1))
    return ${true}
}



function __shell_select() {
    declare -a array_of_commands=("")
    declare -i n=0
    declare -i i=0

    # getting common parameters.
    array_of_commands[i++]=""
    array_of_commands[i++]=""
    array_of_commands[i++]=""

    # select 1
    #        2
    #        3
    # asking arguments.
    select s in ${array_of_commands[@]}
        do
        action="${s}"
        break
    done

    return ${true}
}



# ################################
# Arguments: any type ($1 -- $9)
# Return:    status_code
# ################################
function __list() {
    array_of_parameter[n++]="/usr/share/emacs/"
```

```
    ${F}echo ""
    ${F}iput "${command} ${array_of_parameter[@]:-}"
    ${F}echo ""
    ${command} ${array_of_parameter[@]:-}
    status_code=$?
    return ${status_code}
}



function __gui() {
  #    array_of_parameter[n++]="--directory /usr/share/emacs/"
    # /usr/share/emacs/site-lisp/site-start.el
    # (require 'site-gentoo)
#    array_of_parameter[n++]="--no-site-file"
    # ~/.emacs.el
#    array_of_parameter[n++]="--no-init-file"
    # run
#    array_of_parameter[n++]="--script ~/foo.el"
    # find-file/visit
#    array_of_parameter[n++]="--file ~/.emacs.elc"
    # load
#    array_of_parameter[n++]="--load ~/.emacs.elc"
    # load
#    array_of_parameter[n++]="--user $USER"
    # A) <empty>
    #    interactive
    # B) --batch
    #    elisp
#    array_of_parameter[n++]="--batch"
    # A) <empty>
    # B) --daemon
#    array_of_parameter[n++]="--daemon"
    # init error calls the debugger
#    array_of_parameter[n++]="--debug-init"

#    array_of_parameter[n++]="--display $DISPLAY"
#    array_of_parameter[n++]="--terminal /dev/tty1"
#    array_of_parameter[n++]="--no-desktop"
#    array_of_parameter[n++]="--no-window-system"
#    array_of_parameter[n++]="--no-shared-memory"
#    array_of_parameter[n++]="--no-splash"
#    array_of_parameter[n++]="--multibyte"
#    array_of_parameter[n++]="--unibyte"


    # Input
    array_of_parameter[n++]="${source}"


    ${F}echo ""
    ${F}iput "${F}exec ${command} ${array_of_parameter[@]:-}"
    ${F}echo ""
    ${F}exec ${command} ${array_of_parameter[@]:-}
    status_code=$?
    return ${status_code}
}



function __cui() {
  #    array_of_parameter[n++]="--directory /usr/share/emacs/"
    # /usr/share/emacs/site-lisp/site-start.el
    # (require 'site-gentoo)
#    array_of_parameter[n++]="--no-site-file"
    # ~/.emacs.el
```

```
#    array_of_parameter[n++]="--no-init-file"
    # run
#    array_of_parameter[n++]="--script ~/foo.el"
    # find-file/visit
#    array_of_parameter[n++]="--file ~/.emacs.elc"
    # load
#    array_of_parameter[n++]="--load ~/.emacs.elc"
    # load
#    array_of_parameter[n++]="--user $USER"
    # A) <empty>
    #    interactive
    # B) --batch
    #    elisp
#    array_of_parameter[n++]="--batch"
    # A) <empty>
    # B) --daemon
#    array_of_parameter[n++]="--daemon"
    # init error calls the debugger
#    array_of_parameter[n++]="--debug-init"

#    array_of_parameter[n++]="--display $DISPLAY"
#    array_of_parameter[n++]="--terminal /dev/tty1"
#    array_of_parameter[n++]="--no-desktop"
#    array_of_parameter[n++]="--no-window-system"
#    array_of_parameter[n++]="--no-shared-memory"
#    array_of_parameter[n++]="--no-splash"
#    array_of_parameter[n++]="--multibyte"
#    array_of_parameter[n++]="--unibyte"


    # Input
    array_of_parameter[n++]="${source}"


    ${F}echo ""
    ${F}iput "${F}exec ${command} ${array_of_parameter[@]:-}"
    ${F}echo ""
    ${F}exec ${command} ${array_of_parameter[@]:-}
    status_code=$?
    return ${status_code}
}



function __stub() {
    array_of_parameter[n++]="--alternate-editor=${UL}nano.sh"
    # $EMACS_SERVER_FILE
#    array_of_parameter[n++]="--server-file="
    # /tmp/.private/<USER>/emacs<UID>/server
#    array_of_parameter[n++]="--socket-name=/var/run/emacs-$$"
    # a) <empty>
    #    daemon --- single user
    # b) --tty
    #    Not sharing the frame!!  Use app-misc/tmux.
    #    daemon -+- multi user (CUI)
    #           +- multi user (CUI)
    #           +- multi user (CUI)
    # c) --create-frame
    #    Not sharing the frame!!  Use app-misc/tmux.
    #    daemon -+- multi user (GUI)
    #           +- multi user (GUI)
    #           +- multi user (GUI)
#    array_of_parameter[n++]="--tty"
#    array_of_parameter[n++]="--create-frame"
#    array_of_parameter[n++]="--eval"
#    array_of_parameter[n++]="--no-wait"
```

```
    # Input
    array_of_parameter[n++]="${source:-~/.emacs.el}"


    ${F}echo ""
    ${F}iput "${F}exec ${command} ${array_of_parameter[@]:-}"
    ${F}echo ""
    ${F}exec ${command} ${array_of_parameter[@]:-}
    status_code=$?
    return ${status_code}
}



function __signal() {
    array_of_parameter[n++]="command=emacs"


    ${F}echo ""
    ${F}iput "${command} ${array_of_parameter[@]:-}"
    ${F}echo ""
    ${command} ${array_of_parameter[@]:-}
    status_code=$?
    return ${status_code}
}



# ################################
# Arguments: any type ($1 -- $9)
# Return:    status_code
# ################################
function __help() {
    ${B}cat <<EOF
Usage: emacs.sh list
  or:           detach
  or:           attach [FILE...]
  or:           signal

  or:           cui [FILE...]
  or:           gui [FILE...]

::Key Binding::
C-b/n/p/f                   ... move: l/b/t/r
M-/C-a/v/v/e                ... move: L/B/T/R
C-c C-n/p                   ... move: #if
M-a/e                       ... nove: syntax
M-C-n/p                     ... nove: block
C-x C-f /user@host:/path    ... e:dired
C-x C-f <path>              ... read
C-x C-w <path>              ... write
C-x C-c                     ... exit
C-x r l                     ... select: bookmark
C-x C-b                     ... select: buffer
C-x 0  OR  C-x 1            ... select: window
C-u <level> C-x $           ... fold
M-C-\                       ... indent
M-;                         ... comment
C-q <any_key>               ... as-is
C-c C-j                     ... e:skk
C-g                         ... cancel
C-x u                       ... undo
C-k                         ... kill
C-y                         ... yank
<region> C-w                ... cut
```

```
<region> M-w               ... copy
C-y M-y...                 ... paste
M-C-r <search>             ... search
M-C-s <search>
  Prev: C-r
  Next: C-s
M-C-% <search> <replace>   ... replace
  Yes:    y
  No:     n
  Edit:   C-r
  Return: M-C-r
Tab                        ... completion
M-Space                    ... abbrev
M-/                        ... dabbrev
M-Tab                      ... e:ispell
M-.                        ... etags
<region> C-c r             ... e:lookup
EOF
    ${F}exit ${false}
}



function __version() {
    ${B}cat <<EOF
::Requirements::
"""""""""""""""""
app-shells/bash-3.2
# a) coreutils + util-linux + find + grep + sed + ...
# b) busybox
sys-apps/coreutils-8.5
sys-apps/util-linux-2.18
# a) nano  ...  1587kiB
# b) vim   ...  9694kiB
# c) emacs ... 40752kiB
virtual/editor-0
app-admin/eselect-emacs-1.13
app-emacs/emacs-common-gentoo-1.2
app-editors/emacs-24.0.91

::ChangeLog::
"""""""""""""
[2012-10-31] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* library_message.sh, library_string.sh, library_math.sh, library_datetime.sh, library_path.sh, library_trap.sh, library_ass
[2012-06-10] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-editors/emacs-24.1
[2012-04-22] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* sys-apps/busybox-1.20  (2.10MiB)
  sys-apps/busybox-1.00  (1.10MiB)
  sys-apps/busybox-0.60  (0.60MiB)
[2011-05-30] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* Bugfix: B=""
[2011-02-15] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* Bugfix: status_code
[2009-07-30] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-editors/emacs-23.1
[2009-11-01] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-shells/bash-3.2  (2.41MiB)
  app-shells/bash-4.0  (5.94MiB)
[2007-06-02] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-editors/emacs-22.1
[2005-08-13] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-shells/bash-2.05
[2002-02-01] blurgle blurgle <blurgle.mailAAAAAAAAAAgmail.com>
* app-editors/emacs-21.1
EOF
    ${F}exit ${false}
```

```
}



#
# ${F}warn "Signal Mapping:"
# ${F}warn "* RETURN: N/A"
# ${F}warn "* DEBUG:  N/A"
# ${F}warn "* EXIT:   N/A"
# ${F}warn "* ERR:    __trace()"
${F}main ${@}
```

. . . . . [code:102.1] [input:102.1 ※] [output:102.1]

```
<< rst2html.sh ./CPP.txt ./CPP.html
```

. . . . . [code:102.1] [input:102.1] [output:102.1 ※]

```
...snip...
```

付録 C　Now enough bashing for today, you the king.

西班牙ラ・マンチャ地方の郷士アロンソ・キハーノは賢く美徳も十分なのですが英雄物語と重ね過ぎ、自らを夢を夢見る愁い顔の騎士とし、何処かに居るはずの理想的女性であるべき姫ドゥルシネーア・デル・トボーソを探す旅を、従者サンチョ・パンサ及び驢馬ロシナンテと始めたのです (「老いる前に燃える様な大恋愛がしたい」の純愛劇に聞こえますが、現在は児童書へ分類されてもこの時代の英雄物語は児童向けでなく数十人の人夫の休憩時間等に利用されるそうで、十五歳の姪を御子ちゃま扱いし十六歳直前のドニャ・クラーラ・デ・ビエドマをおぼこ扱いで五年後の淑女を小馬鹿にする、だけど適年齢の娘は無条件に自分を騎士と扱ってくれると思っている辺りが笑い所なのだと思われます)。

1605 年の前編では、奇妙奇天烈な行動 (風車の巨人へ突撃し、田舎娘を呪いがかけられた御嬢様と呼び、阿弗利加ギニア地方？エチオピア地方？矛盾しまくりミコミコン大王国女王ミコミコーナを疑わず、面白料理の霊薬で腹を下し、床屋の金盥を最強兜代わりにし、罪人の言葉を信じ警官に喧嘩を売る) を続ける主人公をからかってやろうとする悪意が無いとするのなら、どうして思慮分別の有る方々が止めるどころか肩を持つのか疑問に持つ常識人？凡人？が登場します。1615 年の後編では、主人公が百姓娘は百姓娘にしか見えないと恋に恋しなくなってしまうのです (元々本作品自体が他作品の真似事から始まるのですが、著者 Miguel de Cervantes 以外が出筆した数々の Don Quijote 贋作を憎む展開が少し目立つ様になります)。騎士と成ったアロンソ・キハーノの遺言 (早死の理由説明？) で物語は終わります。

果たして夢を夢見る郷士が助けた相手から恩返しどころか略奪されて落ち込む話なのか、騎士道精神は誰のためのものと言う話なのか、それとも人生とは楽しくするものである (Not 人生とは楽しいものである) と言う話なのか……な展開は有りますが、其れが主題では無いのかも知れません。斧状長柄武器や板金鎧が流行ったのは十五世紀十六世紀の話で、既に火薬と細身直刀の時代で有り、加えて英雄物語と違い現実の重装騎兵は数を集中させなければ歩兵の餌食なのも主人公は無視します。1492 年に新大陸の発見後、1520 年にアステカ王国征服 1532 年にインカ帝国征服の西班牙も、1588 年に西班牙無敵艦隊がアルマダ海戦において Tudor 王朝 Elizabeth 女王配下の英国艦隊に撃破されます。西班牙では劇作家ロペ・デ・ベーガが、英国では劇作家ウィリアム・シェークスピアが、そんな新しい世界と古い世界も作品は風刺しています。今の日本で言えば大人が楽しむ新聞の川柳みたいなものなのかも。